

TITLE

PIPELINED ADAPTIVE DECISION FEEDBACK EQUALIZER

BACKGROUND OF THE INVENTION

Field of the Invention

5 The invention relates to a decision feedback equalizer (DFE), and more particularly to a pipelined adaptive decision feedback equalizer (ADFE) that is able to relax the iteration bound in the decision feedback loop (DFL) and maintain signal to noise ration (SNR) equal to that of
10 traditional DFE.

Description of the Related Art

Adaptive decision feedback equalizer (ADFE) using the least mean-squared (LMS) algorithm is a well-known equalization technique for magnetic storage and digital
15 communication. The basic block diagram of traditional ADFE is depicted in Fig. 1, where ADFE is composed of two main finite impulse response (FIR) filters, the feedforward filter (FFF) 102 and the feedback filter (FBF) 104. A signal $x(n)$ received from a channel is input to the ADFE.
20 The outputs from both filters are added by an adder 112 and fed into a slicer 106. The signal output from the slicer 106 is the final equalized data. The basic function of the FFF 102 and the FBF 104 is to cancel the pre-cursor and post-cursor inter-symbol interference (ISI) respectively,
25 while the WUC 108 and WUB 110 in the figure stand for the weight-update blocks for the FFF 102 and the FBF 104. Delay units D represent one-tap delay blocks. Delay units D_1 and

delay units D_2 respectively represent n_1 -tap delay blocks and n_2 -tap delay blocks. In addition, n_1 and n_2 are positive integers. A vector of error values $e(n)$ computed as the difference between the output of the slicer 106 and the
s output of the adder 112 outputs from an adder 114. The vector of error values $e(n)$ is respectively delayed by the delay units D_1 and the delay units D_2 , and then respectively fed back to the WUC 108 and WUB 110 to adapt the tap weights.

10 Basically, the fine-grain pipelining of the ADFE is known to be a difficult problem for high-speed applications. This is due to the decision feedback loop (DFL). According to the Iteration Bound theory, the smallest clock period of ADFE is bounded by the DFL. Thus, the presence of the above
15 adaptation loop makes it even more difficult to achieve pipelining.

Several approaches are proposed to solve the aforementioned problems. For example, pipelining the ADFE can be achieved by pre-computing all possible in DFL to open
20 the DFL. (See K. K. Parhi, "Pipelining in algorithm with quantizer loops," *IEEE Trans. Circ. Syst.*, vol. 38, pp.745-754, July 1991). However, the parallel approach results in large hardware overhead as it transforms a serial algorithm into an equivalent (in the sense of input-output behavior)
25 pipelined algorithm. Another algorithm is proposed in Naresh R. Shanbhag, and Keshab K. Parhi, "Pipelined adaptive DFE architectures using relaxed look-ahead," *IEEE Trans. Signal Processing*, vol. 43, No. 6, pp. 1368-1385, June 1995 (hereinafter Naresh et al), which is referred as PIPEADFE2.

It maintains the functionality instead of input-output behavior using the technique of relaxed look-ahead.

Fig. 2 is a block diagram illustrating a configuration of the PIPEADFE2 as disclosed in Naresh et al. The PIPEADFE2 is composed of two main FIR filters, the feedforward filter (FFF) 202 and the feedback filter (FBF) 204. A signal $x(n)$ received from a channel is input to the PIPEADFE2 and n is a time instance. The signal $x(n)$ is delayed by a delay unit D , and then input to a pre-processing unit (PP) 220. The PP 220 receives coefficients from the WUB 210. The output of the PP 220 and the signal $x(n)$ are added by an adder 216 and fed into the FFF 202.

In addition, the outputs from the FFF 202 and the FBF 204 are delayed by delay units D_1 , and then added by an adder 212 and fed into a slicer 206. The signal output from the slicer 206 is the final equalized data. The basic function of the FFF 202 and the FBF 204 is to cancel the pre-cursor and post-cursor ISI respectively, while the WUC 208 and WUB 210 in the figure stand for the weight-update blocks for the FFF 202 and the FBF 204. $C(n)$ is the vector of FFF coefficients and $D(n)$ is the vector of FBF coefficients.

Moreover, delay units D represent one-tap delay blocks. The delay units D_1 and delay units D_2 respectively represent n_1 -tap delay blocks and n_2 -tap delay blocks, where n_1 and n_2 are positive integers. An error vector $e(n)$ output from an adder 214 defines a vector of error values computed as the difference between the output of the slicer 206 and the output of the adder 212. Then, the error vector $e(n)$ is

respectively fed back to the WUC 208 and WUB 210 to adapt the tap weights.

The algorithm used in the PIPEADFE2 is explained as follows.

5 The channel is assumed as $(1+az^{-1})$ where the post-cursor term is a ($a < 1$), and there is no pre-cursor term. For the traditional ADFE shown in Fig. 1, after the ADFE is converged, the first FBF weight is $(-a)$. The coefficients of the FFF 202 are the inverse transfer function of pre-
10 cursors, but those of the FBF 204 are exactly negative value of post-cursors. However, the datum, to be multiplied by the first weight $(-a)$, is in the DFL, and the circuit will limit the speed.

 In PIPEADFE2, the input signal $x(n)$ is filtered by the
15 PP 220, where the transfer function is $1-az^{-1}$ (put the coefficient into the PP 220, which can be found in Fig.6 or Eq. (24b) in Naresh et al). After combining the transfer function of the channel and the PP 220, the effective transfer function combining PP and the channel is $(1+az^{-1})(1-az^{-1}) = 1-a^2z^{-2}$, where the term z^{-1} disappears. It means
20 the transfer function of FBF 204 is modified from $(-a+0z^{-1})$ to $(0+a^2z^{-1})$. Thus, the first coefficient can be zero to relax the critical path by a tap of delay. After the cut-set transform, the delay can be moved to the output of FBF
25 204. Therefore, the algorithm of PP-ADFE can be derived.

 Secondly, the channel is considered as $(1+az^{-1} + bz^{-2})$ and the PP 220 is increased to 2 taps. The effective transfer function of the PP 220 and the channel is $(1+az^{-1} + bz^{-2})(1-az^{-1} - bz^{-2}) = 1+0z^{-1} -a^2z^{-2} -2abz^{-3} -b^2z^{-4}$. For low
30 pass channel, i.e., $1 > a > b$, the term $(+2abz^{-3} -b^2z^{-4})$ is very

small and the performance is almost the same as traditional ADFE. However, on the other hand, for a specific band pass filter, $1 > b > a$, the duration of FBF 204 should be extended and the term $(-2abz^{-3} - b^2z^{-4})$ is large enough to degrade the
5 convergence speed.

Therefore, even though the DFL problem can be addressed by PIPEADFE2, the output SNR of PIPEADFE2 is channel dependent and will be degraded in non-lowpass type channels.

SUMMARY OF THE INVENTION

10 An object of the present invention is to provide a pipelined adaptive decision feedback equalizer (ADFE). Except for increasing the operation speed of the ADFE, the ADFE is capable of relaxing the iteration bound in the decision feedback loop (DFL) and simultaneously maintain
15 signal to noise ration (SNR) equal to that of traditional DFE.

The present invention provides a pipelined adaptive decision feedback equalizer (DFE). A signal received from a channel is equalized by the pipelined ADFE. The pipelined
20 ADFE comprises the pre-processing unit, a feedforward filter (FFF), an adder, a slicer, a feedback filter (FBF), a delay unit, a feedforward filter weight-update block (WUC), a feedback filter weight-update block (WUB) and a mapping circuit. The pre-processing unit (PP) comprising a
25 plurality of PP coefficients filters the signal received from the channel, and generates a PP output signal to the FFF. The coefficients in PP are calculated by the mapping circuit. On the other hand, the inputs of the MAP circuit are the coefficients of FBF.

The FFF comprising a plurality of adaptive coefficients cancels pre-cursor intersymbol interference (ISI) and outputs a precursor free signal to the adder, which adds both output signals from FBF and FFF respectively. The WUB
5 adapts the filter coefficients to cancel the pre-cursor ISI. The slicer outputs a final decision signal based on the output of the adder. The FBF comprises a plurality of FBF coefficients. The FBF receives the final decision signal from the slicer and generates a FBF output signal. The
10 delay unit receives the FBF output signal and outputs a delayed FBF output signal to the adder. The WUC adapts the coefficients which feed into FBF to cancel the post-cursor ISI. The mapping circuit translates the FBF coefficients by a predetermined method to generate the PP coefficients. At
15 least one element of the set of the PP coefficients is different from the corresponding element of the set of the FBF coefficients.

DESCRIPTION OF THE DRAWINGS

The present invention is herein described by way of
20 exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

Fig. 1 is a block diagram illustrating a configuration of a traditional DFE.

25 Fig. 2 is a block diagram illustrating a configuration of a pipelined ADFE as disclosed in Naresh et al.

Fig. 3 is a block diagram illustrating a configuration of a pipelined ADFE according to the embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 3 is a block diagram illustrating a configuration of a pipelined ADFE according to the embodiment of the present invention.

5 The pipelined ADFE is composed of two main FIR filters, the feedforward filter (FFF) 302 and the feedback filter (FBF) 304. A signal $x(n)$ received from a channel is input to the pipelined ADFE and n is a time instance. After being delayed by a delay unit D , the signal $x(n)$ is input to a
10 pre-processing unit (PP) 320. The PP 320 comprises at least one coefficient (not shown in Fig. 3) generated from a map circuit 320 and a delay unit (not shown in Fig. 3). The delay unit in the PP 320 is an m -tap delay block and m is a positive integer. The PP 320 receives updated coefficients
15 from a mapping circuit (MAP) 330. The output of the PP 320 is fed into the FFF 302.

In addition, the output from the FBF 303 is delayed by a delay unit D_1 and then input to an adder 312. The outputs from the FBF 304 and the delay unit D_1 are added by the
20 adder 312. The signal output from the adder 312 is the equalized data, and is fed into a slicer 306. The signal output from the slicer 306 is the final decision data. The basic function of the FFF 302 and the FBF 304 is to cancel the pre-cursor and post-cursor ICI respectively, while the
25 WUC 308 and WUB 310 in the figure stand for the weight-update blocks for the FFF 302 and the FBF 304. $C(n)$ is the vector of FFF coefficients and $D(n)$ is the vector of FBF coefficients.

In this embodiment of the invention, the WUC 308 and WUB 310 use Delay Least-Mean-Square (DLMS) to adapt the tap weights. Delay units D represent one-tap delay blocks. The delay units D_1 , D_2 , D_3 , and D_4 respectively represent n_1 -tap delay blocks, n_2 -tap delay blocks, n_3 -tap delay blocks and n_4 -tap delay blocks, where n_1 , n_2 , n_3 , and n_4 are positive integers. An error vector $e(n)$ output from an adder 314 defines a vector of error values computed as the difference between the output of the slicer 306 and the output of the adder 312. Then, the error vector $e(n)$ is respectively fed back to the WUC 308 and WUB 310 to adapt the tap weights.

The WUB 310 also selects at least one mapping coefficient from the set of the coefficients of the WUB and input the mapping coefficient to the MAP 330. The MAP 330 translates the mapping coefficient by a predetermined method to generate the updated coefficients of the PP 320. At least one element of the set of the mapping coefficient is different from the corresponding element of the set of the filter coefficients of the PP 320.

The PP 320 is coupled before the FFF 302. However, the coefficients of the PP 320 are not directly connected from the coefficients of the WUB 210 as shown in Fig 2. The coefficients of the PP 320 are updated by coefficients output from the MAP 330. After inserting the MAP function, the first m taps of the WUB 310 will decrease. As shown in Fig. 3, the first m taps are moved from the input terminal to the output terminal of the FBF 304 and these weights can be removed to relax DFL.

One example of the MAP function is explained as follows.

The coefficient of the WUB 310 which will be fed into the MAP 330 is assumed as a_i . The coefficient of the PP 320 which is copied from the output of the MAP 330 is assumed as b_i . The relationship between a_i and b_i is,

$$(1 - \sum_{i=1}^M a_i x^i)(1 + \sum_{j=1}^N b_j x^j) = 1 + \sum_{k=1}^{M+N} c_k x^k, \exists c_k = 0 \text{ if } 0 < k < n_1, \quad (1)$$

where M is the number of the coefficients of the PP 320, N is the number of the coefficients of the WUB 310, and M, N, i, j and k are positive integer. Thus, for a given m , we can find a MAP function to translate the coefficients of the PP from the WUB. It should be noted that, $(m+1)$ is the speedup factor of the pipelined ADFE. Even though the MAP 330 is the feedback circuit of the pipelined ADFE, the MAP 330 is not a time critical part relative to the slowly converged WUB. Therefore, the MAP 330 can be easily pipelined to relax the path delay.

In addition, when $M=1$ and $N=1$, the output of the MAP 330 is the same as the output of the WUB 210 shown in Fig. 2.

The principle of the MAP 330 is explained as follows.

The Z-transform of the post-cursor channel is assumed as

$$H(z) = (1 + \sum_{i=1}^M A_i z^{-i}).$$

After the FBF 304 converged, the final coefficients of the FBF 304 are exactly the negative value of channel coefficients. That is,

$$a_i = -A_i. \quad (2)$$

From Eqs.(1) and (2), the effective channel response joint with the PP 320 and real channel is,

$$H(z)PP(z) = (1 + \sum_{i=1}^M A_i z^{-i})(1 + \sum_{j=1}^N b_j z^{-j}) = 1 + \sum_{k=1}^{M+N} c_k z^{-k}, \exists c_k = 0, 0 < k < n_1$$

where c_k will be the coefficients of the FBF 304. Because c_k are all zeros in the first m taps of the FBF 304, after using the above technique of relaxed look-ahead, the delays can be moved to the output of the FBF 304, which is shown in Fig.3 as delay unit D_1 .

And thus, the present pipelined ADFE is able to relax the iteration bound in DFL and keep the SNR as high as traditional DFE.

The foregoing description of the preferred embodiments of this invention has been presented for purposes of illustration and description. Obvious modifications or variations are possible in light of the above teaching. The embodiments were chosen and described to provide the best illustration of the principles of this invention and its practical application to thereby enable those skilled in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. All such modifications and variations are within the scope of the present invention as determined by the appended claims when interpreted in accordance with the breadth to which they are fairly, legally, and equitably entitled.